

Load native Windows API functions

Three ways to achieve this:

1) Use the Add-Type cmdlet to compile C# code 2) Get reference to private type in .NET framework that calls the method 3) Use reflection to dynamically define a method that calls the Windows API function

Add-Type cmdlet

C# signatures can be retrieved from: <http://www.pinvoke.net/>

Note the "public" keyword, this is required to properly use the function from PowerShell

```
$MethodDefinition = @'
[DllImport("kernel32.dll", CharSet = CharSet.Unicode)]
public static extern bool CopyFile(string lpExistingFileName, string lpNewFileName, bool bFailIfExists);
'@

$Kernel32 = Add-Type -MemberDefinition $MethodDefinition -Name 'Kernel32' -Namespace 'Win32' -PassThru
# You may now call the CopyFile function
# Copy calc.exe to the user's desktop
$Kernel32::CopyFile("$($Env:SystemRoot)\System32\calc.exe", "$($Env:USERPROFILE)\Desktop\calc.exe", $False)
```